

EXHIBIT 13

Application of U.S. Patent No. 11,433,344 to the Accused Phones^{*,**}

* The term “Accused Phones” refers to Samsung’s products (including at least Samsung’s Galaxy smartphones, Galaxy Note smartphones, and Galaxy Fold smartphones), as well as all other products made, used, sold, offered for sale and/or imported by Samsung (as defined in the Complaint) or one of Samsung’s affiliated companies, that have the features shown in this chart, or substantially similar features.

** This claim chart is meant to be illustrative for purposes of meeting Plaintiff’s pleading obligations and should not be construed as limiting or binding.

US. Patent No. 11,433,344 versus Accused Phones

29. A mobile wireless device comprising:

Samsung makes, uses, sells, offers for sale, and/or imports mobile wireless devices such as Galaxy smartphones, and Galaxy Note smartphones, and Galaxy Fold smartphones.



https://www.samsung.com/latin_en/smartphones/galaxy-s24/specs/

one or more radios for providing reception and transmission for each of a Bluetooth wireless protocol and a second wireless protocol;

Samsung devices such as Galaxy phones include multiple radios to include a Bluetooth Low Energy (BLE) radio and other options for the radio that uses the second wireless protocol. These other protocols include cellular protocols (LTE, 5G), WiFi and Bluetooth Classic.

Network/Bearer	Number of SIM	SIM size
	Dual-SIM	Nano-SIM (4FF), Embedded-SIM
	SIM Slot Type SIM 1 + SIM 2 / SIM 1 + eSIM / Dual eSIM	Infra 2G GSM, 3G WCDMA, 4G LTE FDD, 4G LTE TDD, 5G Sub6 FDD, 5G Sub6 TDD
2G GSM GSM850, GSM900, DCS1800, PCS1900		3G UMTS B1(2100), B2(1900), B4(AWS), B5(850), B8(900)
4G FDD LTE B1(2100), B2(1900), B3(1800), B4(AWS), B5(850), B7(2600), B8(900), B12(700), B13(700), B17(700), B18(800), B19(800), B20(800), B25(1900), B26(850), B28(700), B66(AWS-3)		4G TDD LTE B38(2600), B39(1900), B40(2300), B41(2500)
5G FDD Sub6 N1(2100), N2(1900), N3(1800), N5(850), N7(2600), N8(900), N12(700), N20(800), N25(1900), N26(850), N28(700), N66(AWS-3)		5G TDD Sub6 N38(2600), N40(2300), N41(2500), N77(3700), N78(3500)

	<p>Connectivity</p> <p>USB Interface USB Type-C</p> <p>USB Version USB 3.2 Gen 1</p> <p>Location Technology GPS, Glonass, Beidou, Galileo, QZSS</p> <p>Earjack USB Type-C</p> <p>MHL No</p> <p>Wi-Fi 802.11a/b/g/n/ac/ax 2.4GHz+5GHz+6GHz, HE160, MIMO, 1024-QAM</p> <p>Wi-Fi Direct Yes</p> <p>Bluetooth Version Bluetooth v5.3</p> <p>NFC Yes</p> <p>PC Sync. Smart Switch (PC version)</p> <p>https://www.samsung.com/latin_en/smartphones/galaxy-s24/specs/</p>
one or more processors; and	<p>Samsung devices such as Galaxy phones include one or more processors.</p> <p>Processor</p> <p>CPU Speed 3.2GHz, 2.9GHz, 2.6GHz, 1.95GHz</p> <p>CPU Type Deca-Core</p> <p>https://www.samsung.com/latin_en/smartphones/galaxy-s24/specs/</p>
one or more memory devices coupled to the one or more processors, the one or more non-transitory memory devices storing a set of instructions that when executed by the one or more processors cause the one or more processors to perform operations including:	<p>Galaxy devices such as Galaxy phones include one or more memory devices coupled to the one or more processors. The one or more memory devices store a set of instructions that when executed by the one or more processors cause the one or more processors to perform operations as discussed below.</p> <p>Storage/Memory</p> <p>Memory_(GB) 8</p> <p>Storage (GB) 256</p> <p>Available Storage (GB) 231.2</p> <p>https://www.samsung.com/latin_en/smartphones/galaxy-s24/specs/</p>
receiving from a beacon transmitter,	<p>Galaxy devices such as Galaxy phones receive from a beacon transmitter, using the Bluetooth wireless protocol, a first plurality of beacon transmissions during a first time period. Each of the beacon transmissions includes (a) a MAC address (<i>e.g.</i>, an AdvA field in the BLE advertising</p>

<p>using the Bluetooth wireless protocol, a first plurality of beacon transmissions during a first time period, each of the beacon transmissions including (a) a MAC address, (b) a unique identifier, wherein the first unique identifier does not directly identify a location of the beacon transmitter, and (c) a beacon service identifier, wherein the beacon service identifier of each respective beacon transmission identifies a wireless beacon service as opposed to identifying a communication network or a device accessible using the Bluetooth wireless protocol;</p>	<p>protocol data unit), (b) a unique identifier, wherein the first unique identifier does not directly identify a location of the beacon transmitter (e.g., Account Key Data sent via BLE during non-discoverable mode from the peripheral device to the Samsung device), and (c) a beacon service identifier (e.g., the serviceId, alone or with additional data from the BLE advertisements), wherein the beacon service identifier of each respective beacon transmission identifies a wireless beacon service as opposed to identifying a communication network or a device accessible using the Bluetooth wireless protocol.</p> <p>“The Provider is a device that is advertising its presence and readiness to pair (e.g. a discoverable pair of headphones)”</p> <p>https://developers.google.com/nearby/fast-pair/specifications/configuration</p> <p>“Fast Pair uses Bluetooth Low Energy (BLE) technology to communicate with your phone. BLE is a wireless protocol that consumes very little power and has a long range. It also has a high level of security, as it encrypts the data that is transmitted between devices. Fast Pair uses BLE to send a unique identifier from your device to your phone, which then contacts Google's servers to get more information about the device, such as its name and icon. This way, you can easily recognize your device on your phone screen and pair it with a single tap.”</p> <p>https://communityin.oppo.com/thread/1301071443236225027</p> <p>“The Google Fast Pair Service (GFPS) utilizes Bluetooth Low Energy (BLE) to discover nearby Bluetooth devices without using significant phone battery, enabling “magical” scenarios based on device proximity.” (...) GFPS is aimed at facilitating the pairing of Bluetooth and BLE devices, such as speakers, headphones, car kits, mice and keyboards, with as little user interaction required as possible.”</p> <p>https://developers.google.com/nearby/fast-pair/specifications/introduction</p> <p>Model Registration</p> <p>All Provider models must be registered with Google before they will work with Fast Pair. After registration, Google will distribute a Model ID and Anti-Spoofing Public/Private Key Pair. The information provided during registration is used in the pairing suggestion presented to the user, and may be used in other UX.</p> <div data-bbox="662 1472 1089 1587" data-label="Image"> </div> <p><i>Example pairing suggestion (subject to change)</i></p> <p>https://developers.google.com/nearby/fast-pair/specifications/service/modelregistration</p> <p>“Provider Advertising signal - Advertising: When discoverable - When the Provider device is BR/EDR discoverable (that is, in pairing mode), it shall advertise Fast Pair Model ID Data over BLE, and the BLE address shall not be rotated.”</p>
--	---

Advertising payload: Fast Pair Model ID Data

The advertisement shall contain the Service Data data type, *ibid.*, § 1.11. The UUID shall be the Fast Pair Service UUID of 0xFE2C. The service data shall contain the following:

Octet	Data type	Description	Value
0-2	uint24	24-bit model ID	varies

<https://developers.google.com/nearby/fast-pair/specifications/service/provider>

“Advertising: When not discoverable

When not discoverable (that is, not in pairing mode), the Provider device shall advertise Fast Pair Account Data, using the following guidelines.

Advertising the account data allows Seekers nearby to recognize when a provider belongs to their account and initiate pairing without having to force the provider back into pairing mode first, which is a common cause for user complaint.”

Advertising payload: Fast Pair Account Data

The advertisement shall contain the Service Data data type, *ibid.*, § 1.11. The UUID shall be the Fast Pair Service UUID of 0xFE2C. The service data shall contain the following:

Octet	Data type	Description	Value
0	uint8	Version and flags 0bVVVFFFF <ul style="list-style-type: none">V = versionF = flags	0x00 (reserved for future use)
1 - varies		Account Key Data	varies or 0x00 if the Account Key List is empty

See also <https://developers.google.com/nearby/fast-pair/specifications/introduction>, and <https://www.slashgear.com/android-fast-pair-new-features-google-pixel-buds-2-27618482/>

2.3.1 Advertising PDUs

The following advertising physical channel PDU Types are called advertising PDUs:

- ADV_IND
- ADV_DIRECT_IND
- ADV_NONCONN_IND
- ADV_SCAN_IND
- ADV_EXT_IND
- AUX_ADV_IND
- AUX_SYNC_IND
- AUX_CHAIN_IND

These PDUs are sent by the Link Layer in the Advertising state and received by a Link Layer in the Scanning state or Initiating state. The ADV_IND, ADV_DIRECT_IND, ADV_NONCONN_IND, and ADV_SCAN_IND PDUs are called "legacy advertising PDUs". The ADV_EXT_IND, AUX_ADV_IND, AUX_SYNC_IND, and AUX_CHAIN_IND PDUs are called "extended advertising PDUs". Advertising events using legacy advertising PDUs are called "legacy advertising events".

2.3.1.1 ADV_IND

The Payload field of the ADV_IND PDU is shown in Figure 2.6. The PDU shall be used in connectable and scannable undirected advertising events. The TxAdd in the advertising physical channel PDU header indicates whether the advertiser's address in the AdvA field is public (TxAdd = 0) or random (TxAdd = 1). The ChSel field in the advertising physical channel PDU header shall be set to 1 if the advertiser supports the LE Channel Selection Algorithm #2 feature (see Section 4.5.8.3).

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.6: ADV_IND PDU payload

The Payload consists of AdvA and AdvData fields. The AdvA field shall contain the advertiser's public or random device address as indicated by TxAdd. The AdvData field, if not empty, shall contain Advertising Data from the advertiser's Host.

Bluetooth Core Specification v5.3, available at

<https://www.bluetooth.com/specifications/specs/core-specification-5-3/>

4.4.2 Advertising state

The Link Layer shall enter the Advertising state when directed by the Host. When placed in the Advertising state, the Link Layer shall send advertising PDUs (see Section 2.3.1) in advertising events, periodic advertising events, or both.

Bluetooth Core Specification v5.3, available at

<https://www.bluetooth.com/specifications/specs/core-specification-5-3/>

“To prevent tracking, BLE advertising shall use the random resolvable private address (RPA). The address shall be rotated at minimum every 15 minutes while the device is actively advertising, and every time the state changes from not advertising to advertising. A randomized offset should be used to alter the address randomization interval.”

<https://developers.google.com/nearby/fast-pair/specifications/configuration>

The **Account Key Data** contains:

Octet	Data type	Description	Value
0	uint8	Field length and type 0bLLLLTTTT • L = length of account key filter in bytes • T = type	0bLLLL0000 • length = 0bLLLL = <i>varies</i> • type = 0b0000 (show UI indication) or 0b0010 (hide UI indication), Account Key Filter
1 - s		Account Key Filter	<i>varies</i>
s + 1	uint8	Field length and type 0bLLLLTTTT • L = length in bytes • T = type	0b00100001 • length = 0b0010 = 2 • type = 0b0001, Salt
s + 2 - s + 3	uint16	Salt	<i>varies</i>

Salt field

The **salt** is a random value that is appended to account keys when building the bloom filter. This **salt** should be regenerated every time the RPA is updated for the Provider to avoid tracking across address rotation.

To generate the Account Key Filter using the **salt**:

1. Generate a random 2-byte S. Note that there is no "endianness" to this value, since there is no more or less significant byte — don't alter the byte order.
2. Use the 2-byte S as the **Salt**.
3. In the advertised Fast Pair Account Data, include the generated filter in the Account Key Filter field, and S in the **Salt** field.

<https://developers.google.com/nearby/fast-pair/specifications/service/provider>

“The Provider shall allocate space to store a persisted list of 128-bit Account Keys. Each Account Key allows the Provider to be recognized as belonging to a certain user account. The list must be able to store at least five keys (that is, there must be at least 80 bytes of space dedicated to this list). (...) see the Account Key Filter section for more information.”

<https://developers.google.com/nearby/fast-pair/specifications/configuration>

“The advertised Account Key Filter allows a Seeker to quickly check whether a Provider might possess a certain account key (with a low false-positive probability, on average much less than 0.5%), before further interactions.”

b. Hash V using SHA256, obtaining a 32-byte value $H = \{H_0, \dots, H_{31}\}$.

```
uint8_t H[32] = {0};
SHA256_hash_function(V, H);
```

<https://developers.google.com/nearby/fast-pair/specifications/service/provider#AccountKeyFilter>

Fast Pair Service

The Fast Pair Provider shall have the following GATT service.

Service	UUID
Fast Pair Service	0xFE2C

This service shall have the following characteristics.

Fast Pair Service characteristic	Encrypted	Permissions	UUID
Model ID	No	Read	FE2C1233-8366-4814-8EB0-01DE32100BEA
Key-based Pairing	No	Write and notify	FE2C1234-8366-4814-8EB0-01DE32100BEA
Passkey	No	Write and notify	FE2C1235-8366-4814-8EB0-01DE32100BEA
Account Key	No	Write	FE2C1236-8366-4814-8EB0-01DE32100BEA

<https://developers.google.com/nearby/fast-pair/specifications/characteristics#KeyBasedPairing>

“Besides the seamless connections, Fast Pair also stores details about the Bluetooth accessory on to your Google ID, so every device running that Google account can automatically identify the accessory. After being added to your Google account, the accessory also appears among other devices in Google's Find My Device app, which can be used to track the accessory's whereabouts or details of the last Android device connected in case you lose it. The feature also allows the automatic transitioning from one connected Android device to another if the Fast Pair-enabled accessory supports multiple simultaneous connections.”

<https://www.slashgear.com/1454224/google-fast-pair-service-android-how-to/>

See also <https://developers.google.com/nearby/fast-pair/specifications/introduction>, and <https://www.slashgear.com/android-fast-pair-new-features-google-pixel-buds-2-27618482/>

	<p>1. Preparation for Certification</p> <p>1.1 Definition</p> <ul style="list-style-type: none"> • Initial pairing is the pairing between a Fast Pair compliant device and a phone where user's Google Account is logged-in at the very first time, and the phone detects the advertisement from the device and recognizes model ID of the device. Then the user follows the instructions shown in the popup notification to connect the device. (In this guideline, 'device' means the Bluetooth headset or speaker instead of a reference phone) • Subsequent pairing is the later pairing between this previously initial-paired device and the other phone where the same Google Account is logged-in, and the phone detects the advertisement and recognizes the account key from the device. Then the user follows the instructions shown in the notification to connect the device. <p>https://developers.google.com/nearby/fast-pair/fast-pair-certification-guideline</p>
receiving from one or more servers, using the second wireless protocol, stored information related to a entity or object associated with a first unique identifier;	<p>Galaxy devices such as Galaxy phones receive from one or more servers, using the second wireless protocol, stored information related to a entity or object associated with a first unique identifier.</p> <p>“GFPS can also display your accessory's battery levels in a persistent notification, aid in passing off audio sourcing (if your audio gear supports multipoint) to a different device as needed, provide an interface on your phone to control certain features on your ear buds like ANC, and helps you locate your accessory by either ringing an alarm through them or showing their last known location while paired.”</p> <p>https://www.pocket-lint.com/what-is-google-fast-pair-and-how-does-it-work/</p> <p>“Case 2: The pre-shared key is one of the account keys. The Provider is usually not in pairing mode. (But this is not a requirement—The Provider should support using an account key even when in pairing mode.) The Seeker and Provider each verify that the other is in possession of the account key. Since both cases are extremely similar, except for which pre-shared key is used, they are combined in procedure.”</p> <p>https://developers.google.com/nearby/fast-pair/specifications/characteristics#KeyBasedPairing</p>
selecting one or more unique identifiers from the first plurality of beacon transmissions, by filtering the beacon transmissions which include a particular beacon service	<p>Galaxy devices such as Galaxy phones select one or more unique identifiers from the first plurality of beacon transmissions, by filtering the beacon transmissions which include a particular beacon service identifier indicating that a particular received beacon transmission is associated with the particular wireless beacon service (e.g., by responding to beacons for Nearby Share if the unique identifier received in the Advertisement beacon corresponds to a User ID in the Google Contacts list).</p> <p>“The advertised Account Key Filter allows a Seeker to quickly check whether a Provider might possess a certain account key (with a low false-positive probability, on average much less than 0.5%), before further interactions.”</p> <p>https://developers.google.com/nearby/fast-pair/specifications/service/provider#AccountKeyFilter</p>

identifier indicating that a particular received beacon transmission is associated with the particular wireless beacon service;	
---	--